Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Лабораторная работа №14. Задание №5**

**Курс: «Проектирование реконфигурируемых гибридных вычислительных систем»**

Выполнил студент гр. 3540901/81501 Селиверстов С.А.

(подпись)

Руководитель Антонов А.П.

(подпись)

“\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г.

Санкт – Петербург

2020

Оглавление

[1. Задание 3](#_Toc30378394)

[2. Исходный код 5](#_Toc30378395)

[3. Ход работы 7](#_Toc30378396)

[3.1. Решение 1а 8](#_Toc30378397)

[3.1.1. Директивы 8](#_Toc30378398)

[3.1.2. Моделирование 8](#_Toc30378399)

[3.1.3. Синтез 8](#_Toc30378400)

[3.1.4. C/RTL моделирование 10](#_Toc30378401)

[3.2. Решение 2а 11](#_Toc30378402)

[3.2.1. Директивы 11](#_Toc30378403)

[3.2.2. Синтез 11](#_Toc30378404)

[3.2.3. C/RTL моделирование 13](#_Toc30378405)

[3.3. Решение 3а 14](#_Toc30378406)

[3.3.1. Директивы 14](#_Toc30378407)

[3.3.2. Синтез 14](#_Toc30378408)

[3.3.3. C/RTL моделирование 16](#_Toc30378409)

[3.4. Решение 4а 16](#_Toc30378410)

[3.4.1. Директивы 17](#_Toc30378411)

[3.4.2. Синтез 17](#_Toc30378412)

[3.4.3. C/RTL моделирование 21](#_Toc30378413)

[4. Выводы 21](#_Toc30378414)

# Задание

* Создать проект lab14\_5
* Микросхема: xa7a12tcsg325-1q
* В папке source текст функции pointer\_double
  + *Познакомьтесь с ним.*
* Познакомьтесь с тестом.
* Исследование:
* Solution\_1а
  + Осуществить моделирование
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию ПО УМОЛЧАНИЮ
  + осуществить синтез для:
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * scheduler viewer (выполнить Zoom to Fit)
        + На скриншоте показать Latency
        + На скриншоте показать Initiation Interval
      * resource viewer (выполнить Zoom to Fit)
        + На скриншоте показать Latency
        + На скриншоте показать Initiation Interval
  + Выполнить cosimulation и привести временную диаграмму
    - Обратить внимание на реализацию интерфейсов.
* Solution\_2а
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию КОНВЕЙЕР ДЛЯ ЦИКЛА
  + осуществить синтез
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * scheduler viewer (выполнить Zoom to Fit)
        + На скриншоте показать Latency
        + На скриншоте показать Initiation Interval
      * resource viewer (выполнить Zoom to Fit)
        + На скриншоте показать Latency
        + На скриншоте показать Initiation Interval
  + Выполнить cosimulation и привести временную диаграмму
* Сравнить два решения (solution\_1a и solution\_2a) и сделать выводы
* Solution\_3а
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию INLINE ДЛЯ ФУНКЦИИ
  + осуществить синтез
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * scheduler viewer (выполнить Zoom to Fit)
        + На скриншоте показать Latency
        + На скриншоте показать Initiation Interval
      * resource viewer (выполнить Zoom to Fit)
        + На скриншоте показать Latency
        + На скриншоте показать Initiation Interval
  + Выполнить cosimulation и привести временную диаграмму
* Сравнить два решения (solution\_1a и solution\_2a и solution\_3a) и сделать выводы
* Solution\_4а
  + задать: clock period 10; clock\_uncertainty 0.1
  + установить реализацию ПРЕДЛОЖИТЕ ДЛЯ УВЕЛИЧЕНИЯ ПРИЗВОДИТЕЛЬНОСТИ И ЗАДЕРЖКИ, + ОПТИМИЗАЦИЯ ИНТЕРФЕЙСА + УМЕНЬШЕНИЕ AREA
  + осуществить синтез
    - привести в отчете:
      * performance estimates=>summary (timing, latency)
      * utilization estimates=>summary
      * performance Profile
      * Resource profile
      * scheduler viewer (выполнить Zoom to Fit)
        + На скриншоте показать Latency
        + На скриншоте показать Initiation Interval
      * resource viewer (выполнить Zoom to Fit)
        + На скриншоте показать Latency
        + На скриншоте показать Initiation Interval
  + Выполнить cosimulation и привести временную диаграмму
* Сравнить с полученными ранее решениями: solution\_1a и solution\_2a и solution\_3a и сделать выводы

# Исходный код

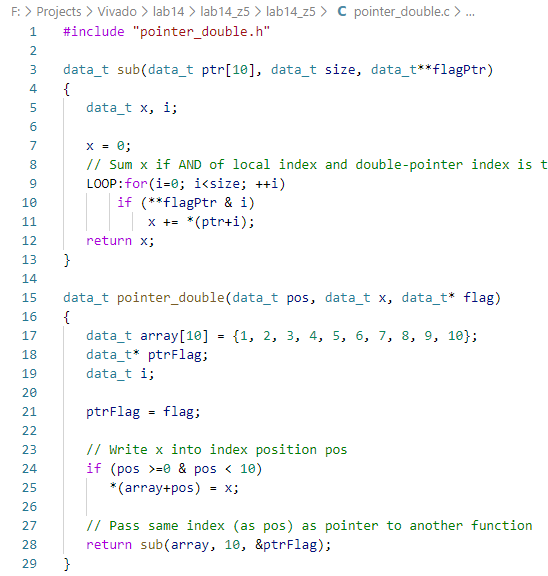


Рисунок . Исходный код устройства

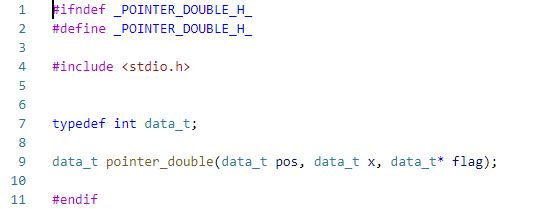


Рисунок . Заголовочный файл

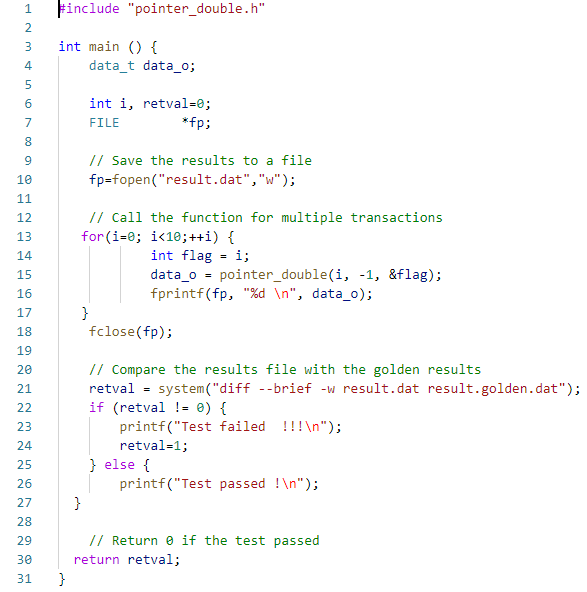


Рисунок . Тест

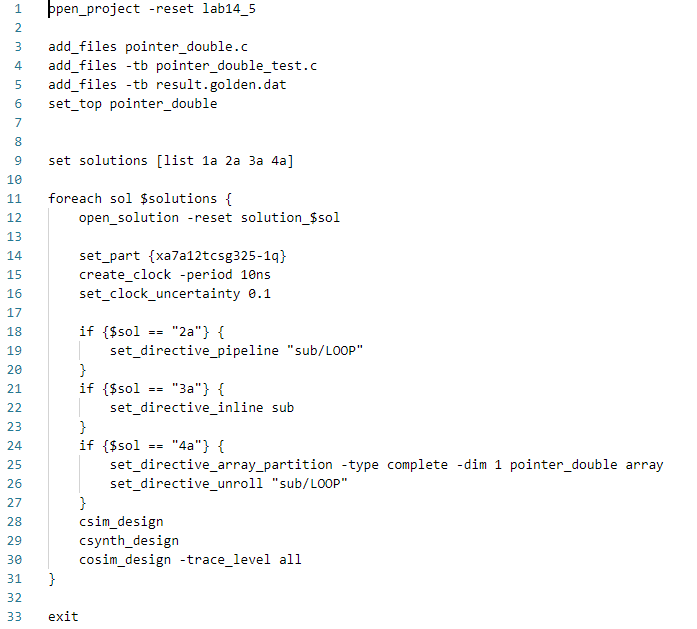


Рисунок . Скрипт

# Ход работы

## Решение 1а

## Директивы

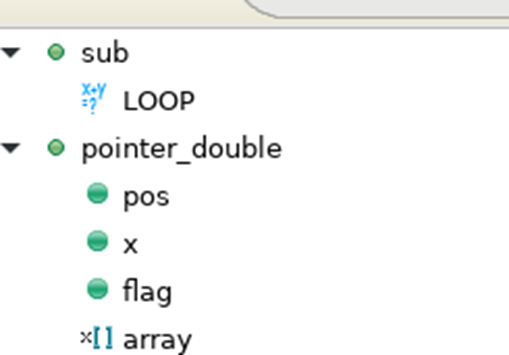


Рисунок . Директивы

## Моделирование

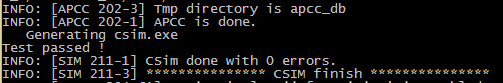


Рисунок . Результаты моделирования

Устройство работает корректно

## Синтез

По оценке производительности видно, что устройство соответствует заданным критериям.

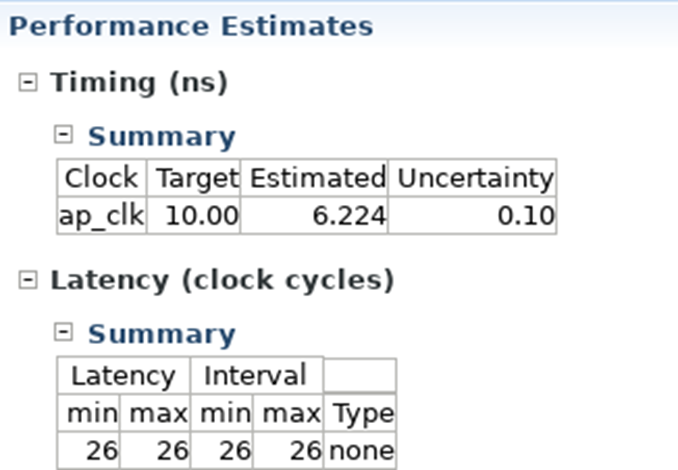


Рисунок . Performance estimates

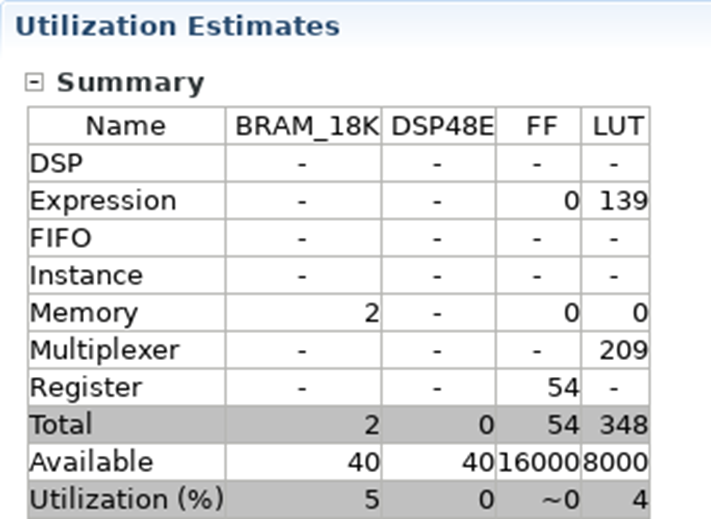


Рисунок . Utilization estimates

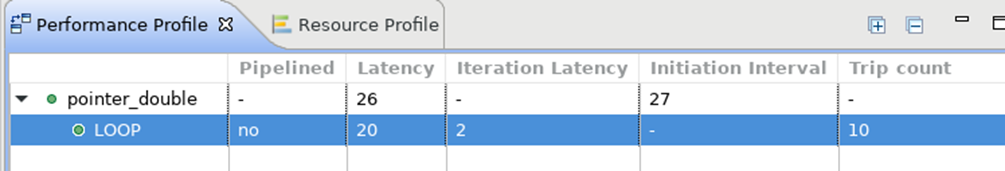


Рисунок . Performance profile

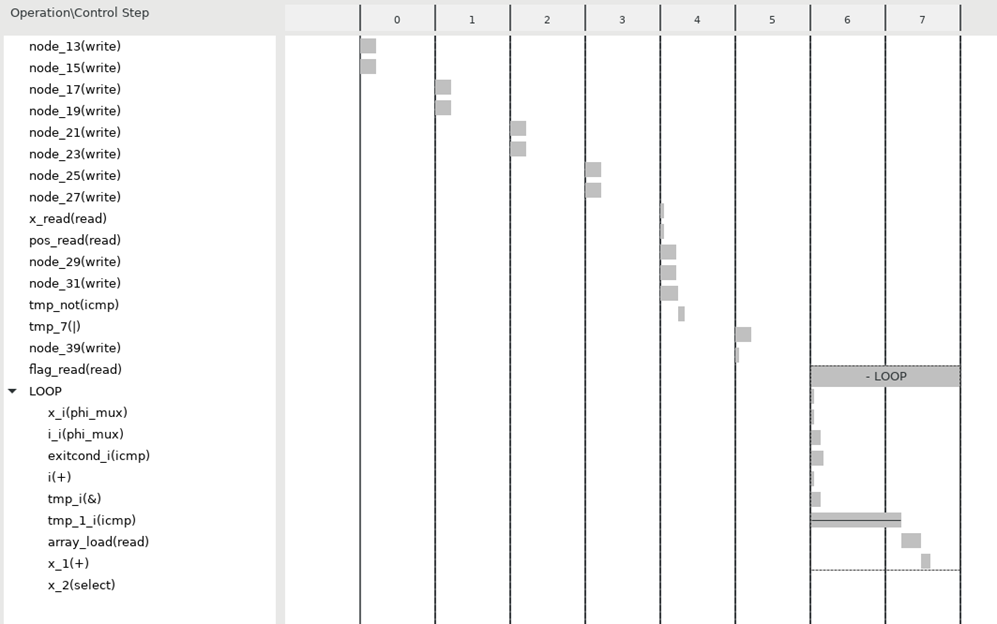


Рисунок . Scheduler viewer

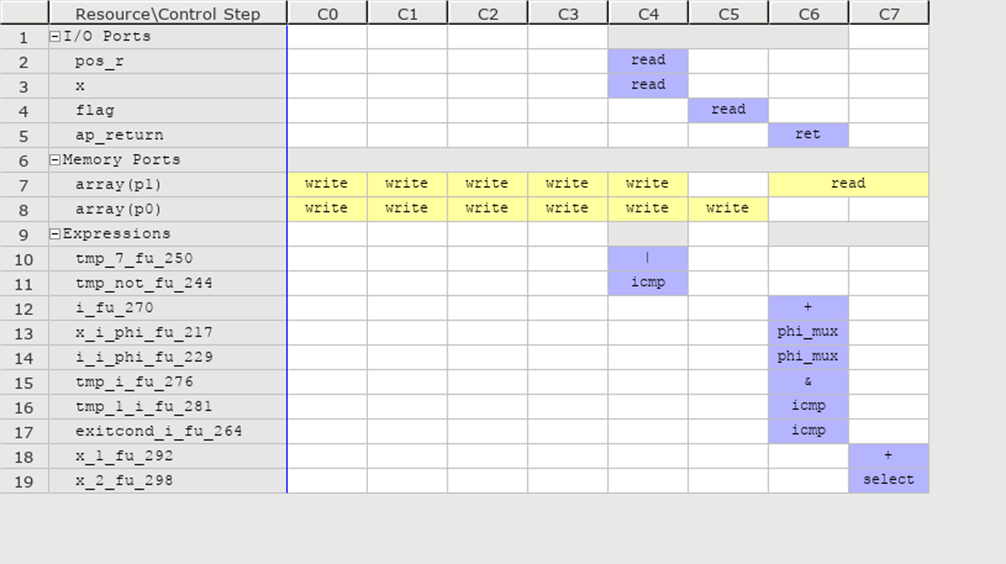


Рисунок . Resource viewer

## C/RTL моделирование

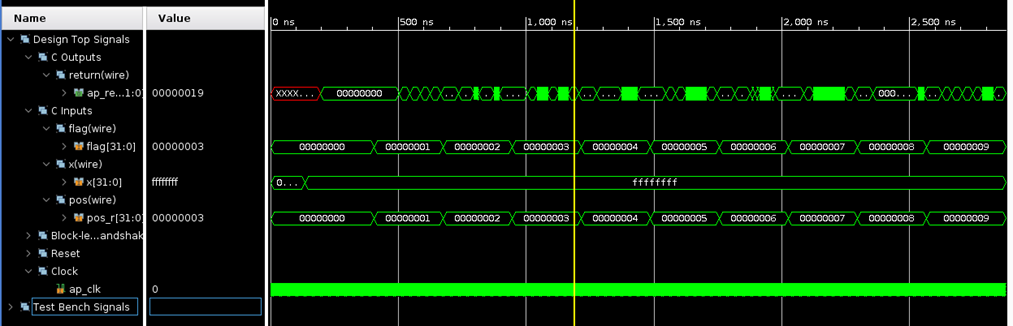


Рисунок . Временная диаграмма

Исходя из временной диаграммы, цикл выполняется 10 раз, на каждую итерацию требуется 2 такта, в итоге Latency = 10\*2 = 20 Кроме цикла в программе присутствую операции записи на которые требуется еще тактов, итоговое значение Latency = 20 + 6 = 26 Значения будут доступны на выходе еще через 1 такт II = Latency + 1 = 27

## Решение 2а

## Директивы

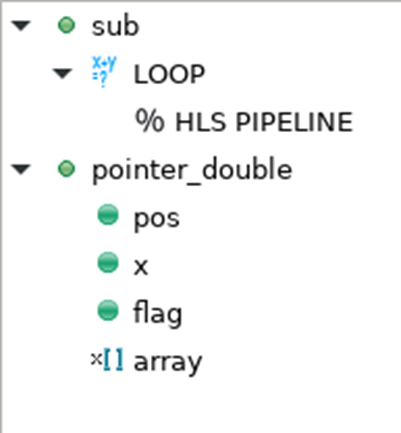


Рисунок . Директивы

## Синтез

По оценке производительности видно, что устройство соответствует заданным критериям.

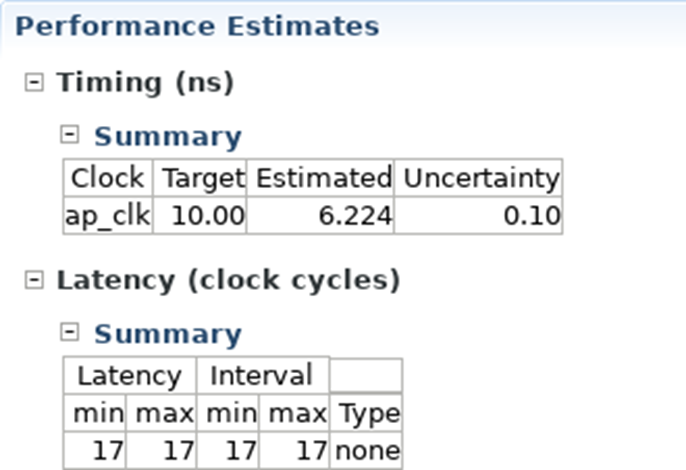


Рисунок .Performance estimates

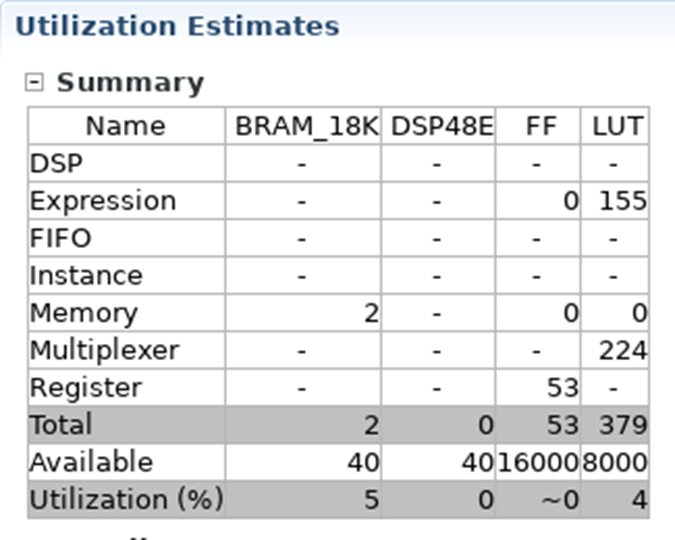


Рисунок . Utilization estimates

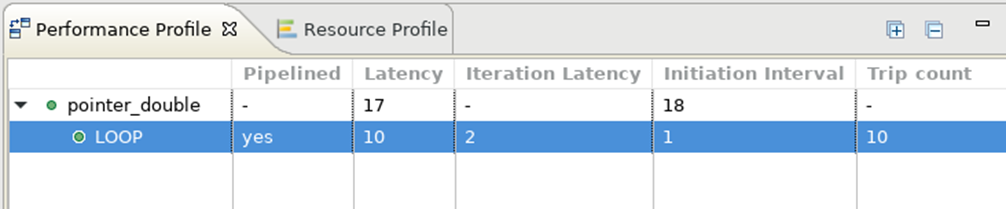


Рисунок . Performance profile



Рисунок . Scheduler viewer

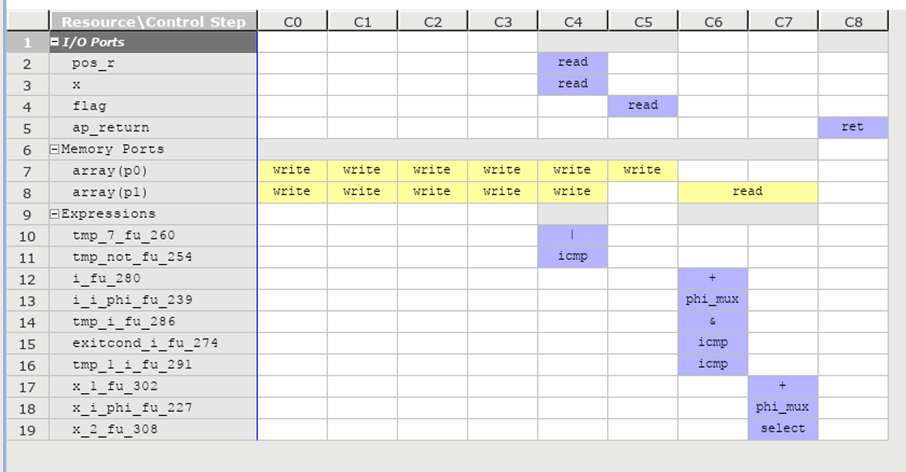


Рисунок . Resource viewer

## C/RTL моделирование

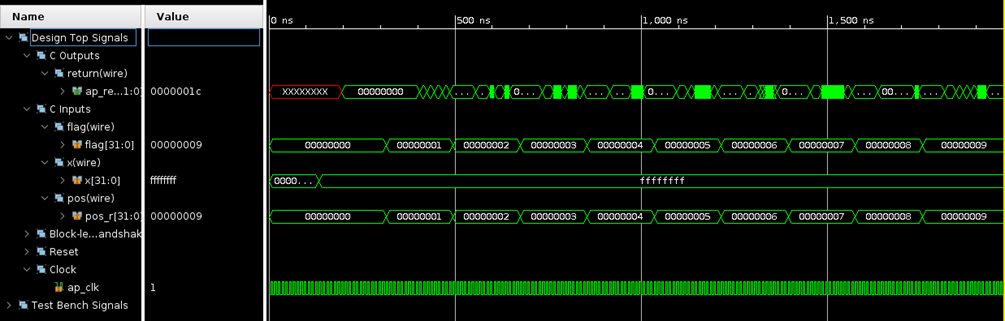


Рисунок . Временная диаграмма

Как видно из временной диаграммы, значение Latency уменьшилось с 26 до 17 за цикл. Это связано с тем что после применения директивы конвейеризации модуль не ждет окончания выполнения всего цикла, а сразу читает следующие данные и подает в функцию.

## Решение 3а

## Директивы

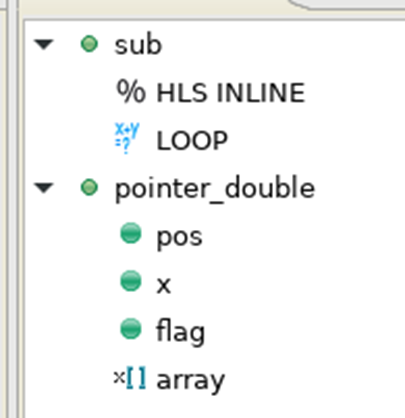


Рисунок . Директивы

## Синтез

По оценке производительности видно, что устройство соответствует заданным критериям

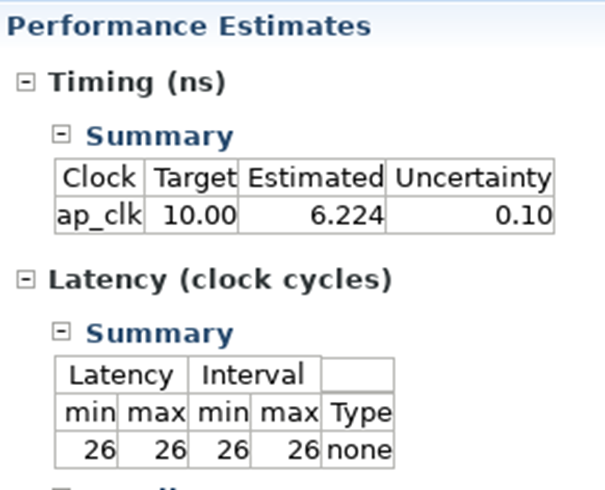


Рисунок . Performance estimates

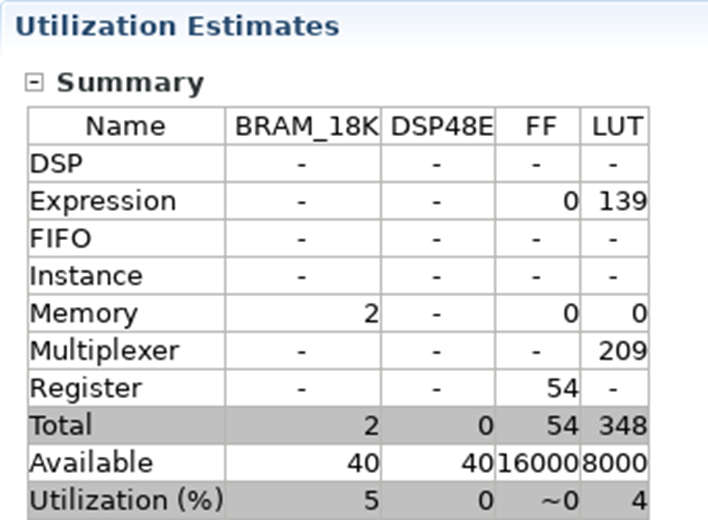


Рисунок . Utilization estimates

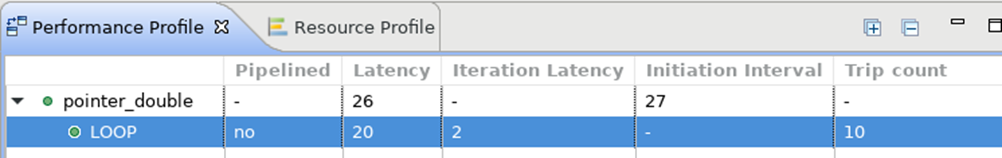


Рисунок . Performance profile



Рисунок . Scheduler viewer

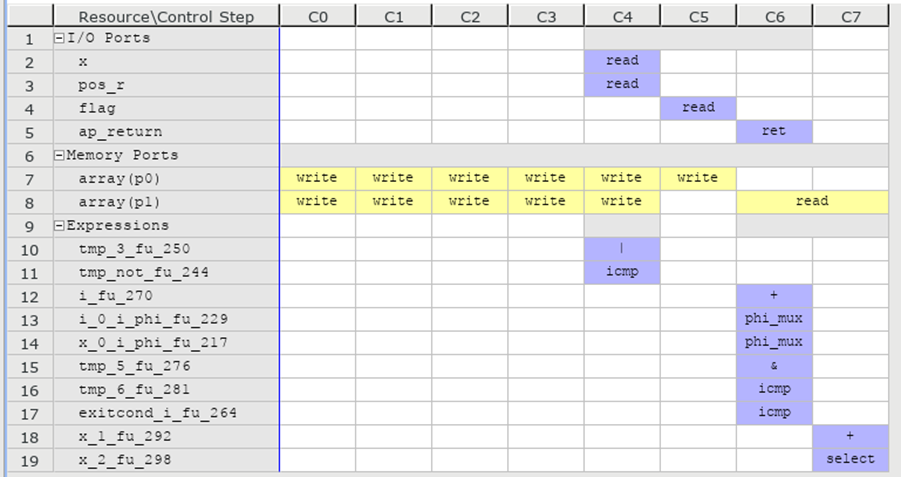


Рисунок . Resource viewerу

## C/RTL моделирование

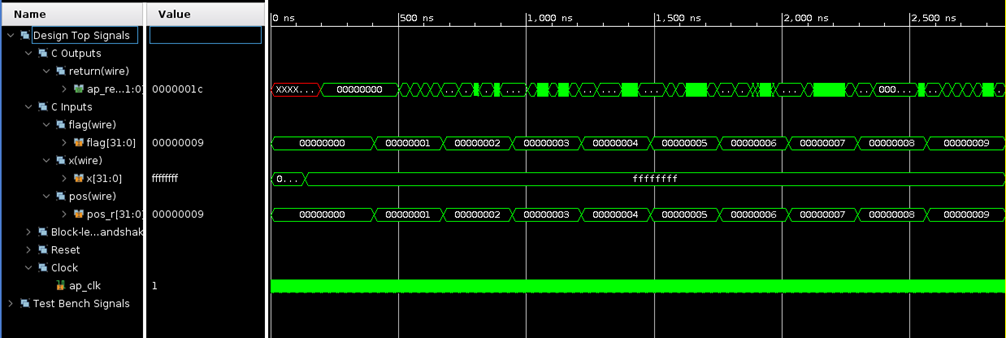


Рисунок . Временная диаграмма

Как видно по результатам, после применения директивы INLINE решение соответствует решению 1 (по-умолчанию), то есть inlining не влияет на производительность данного устройства.

## Решение 4а

Исходя из временных диаграмм полученных ранее, можно сделать вывод, что дольше всего выполняются команды записи и чтения в массив array. Для уменьшения временных задержек можно применить директиву ARRAY\_PARTITION для массива array и директиву UNROLL для цикла.

## Директивы

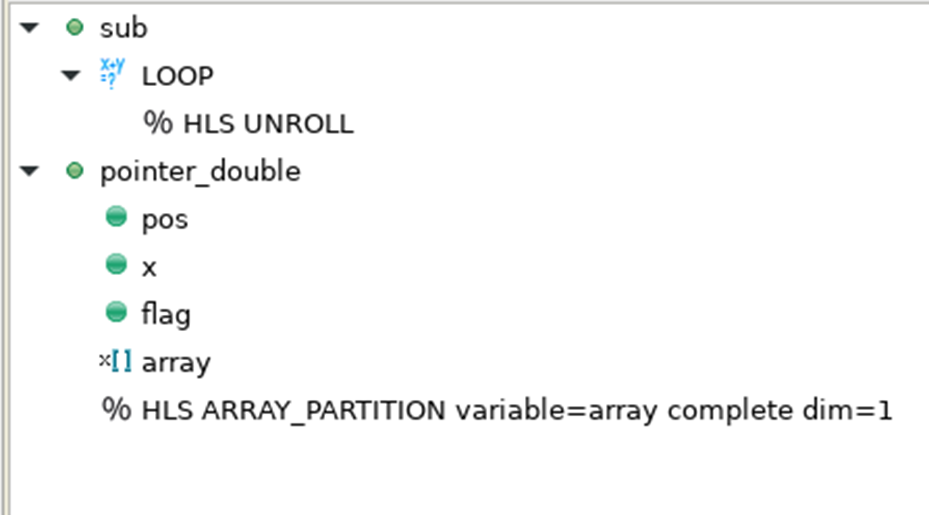


Рисунок . Директивы

## Синтез

По оценке производительности видно, что устройство соответствует заданным критериям.

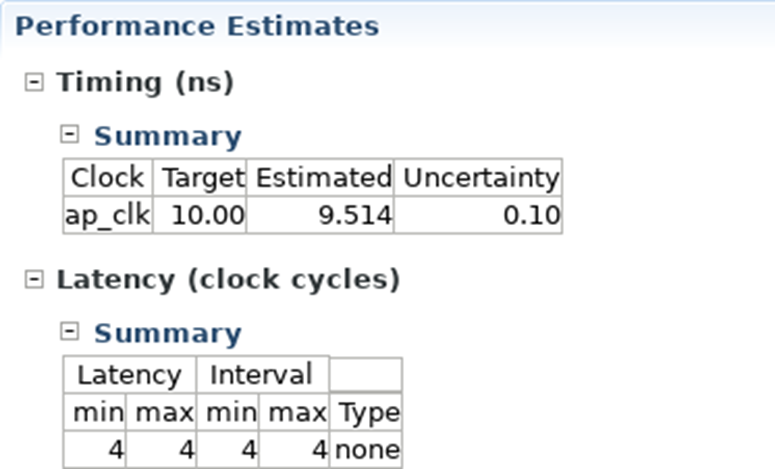


Рисунок . Performance estimates

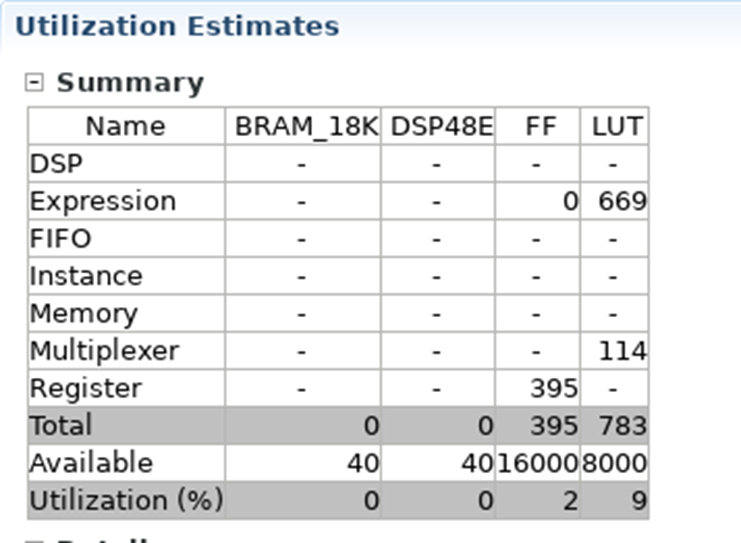


Рисунок . Utilization estimates

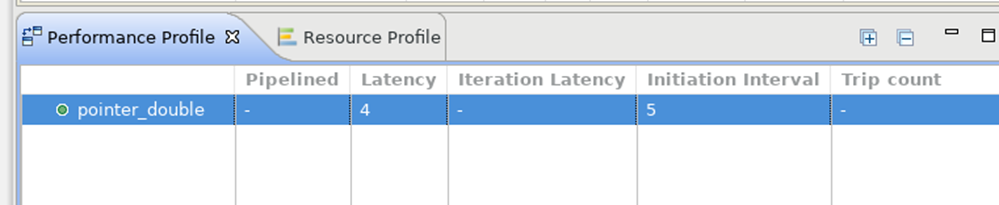


Рисунок . Performance profile

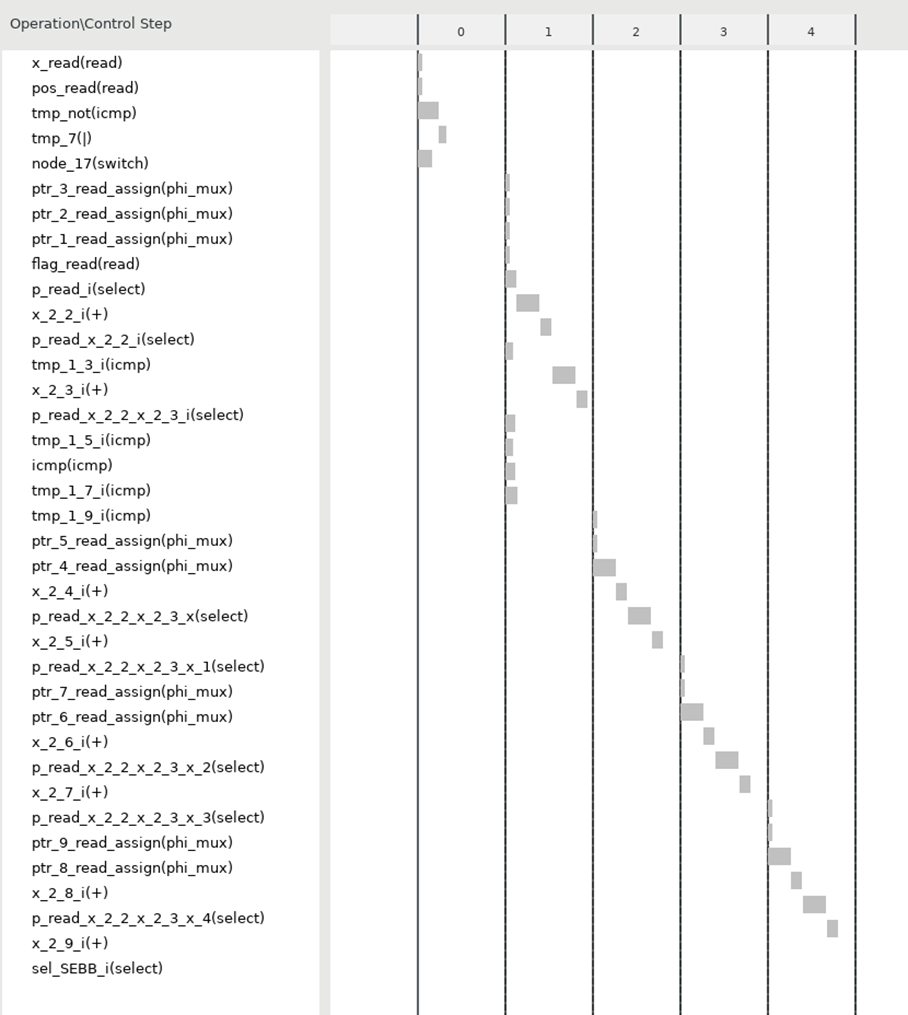


Рисунок . Scheduler viewer

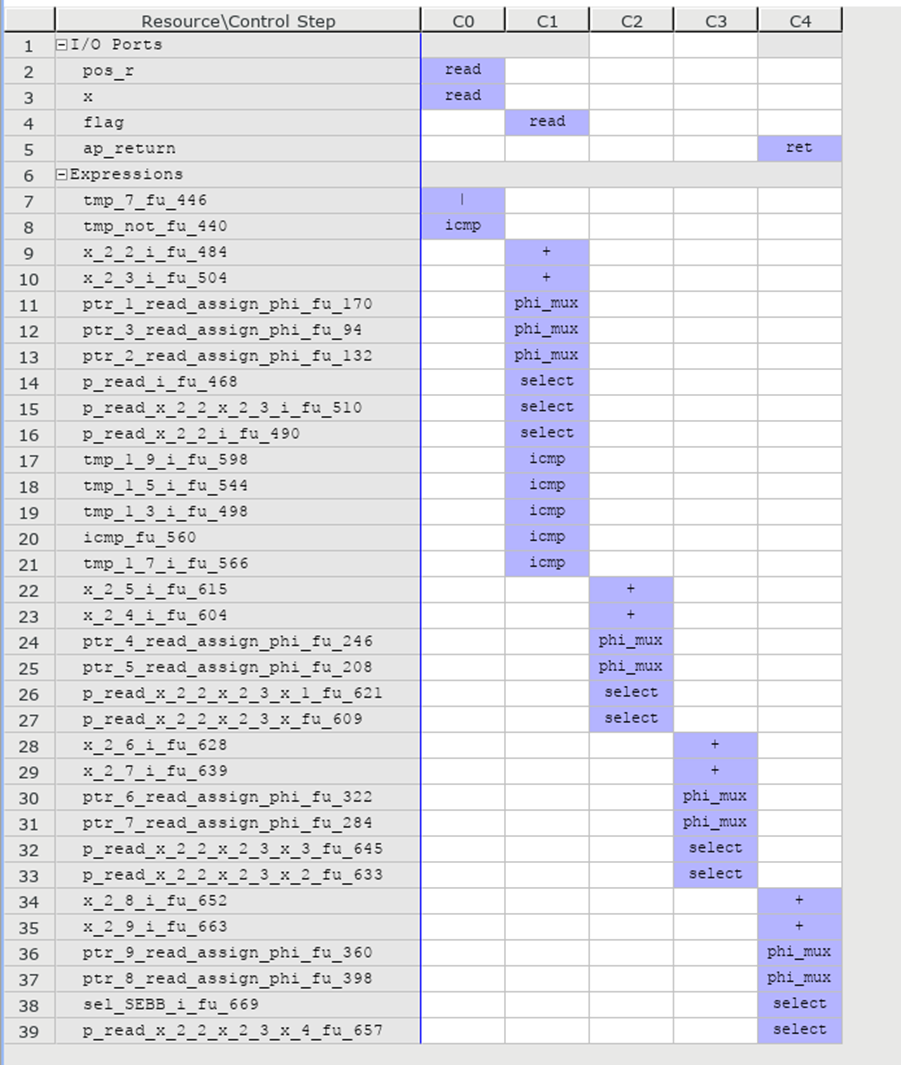


Рисунок . Resource viewer

## C/RTL моделирование

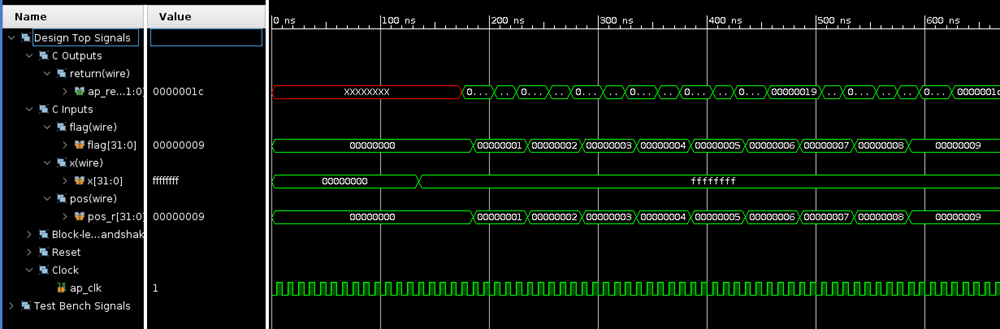
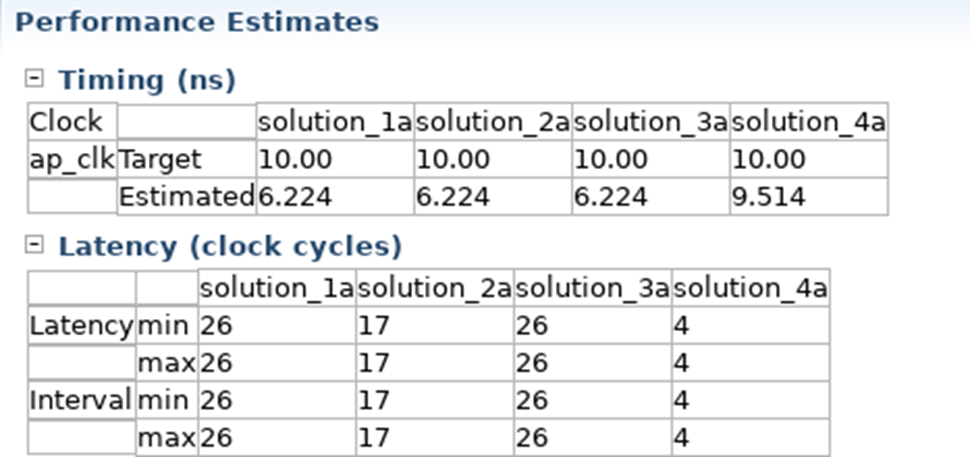


Рисунок . Временная диаграмма

По результатам видно, что в данном решении достигнута наибольшая производительность по сравнению с остальными.

# Выводы



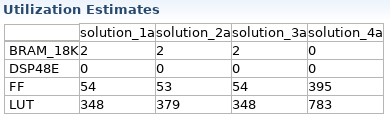


Рисунок . Сравнение производительности и затрат ресурсов

По результатам можно сделать вывод, что применённые директивы улучшают производительность.